MeMAD

Methods for Managing Audiovisual Data

memad.eu
info@memad.eu

Twitter – @memadproject
LinkedIn – MeMAD Project

MeMAD Deliverable

D4.3 Tools and Models for Multimodal, Multilingual and Discourse-Aware Machine Translation

Grant agreement number	780069
Action acronym	MeMAD
Action title	Methods for Managing Audiovisual Data: Combining Automatic Efficiency with Human Accuracy
Funding scheme	H2020–ICT–2016–2017/H2020–ICT–2017–1
Version date of the Annex I against which the assessment will be made	23.6.2020
Start date of the project	1.1.2018
Due date of the deliverable	31.03.2020
Actual date of submission	16.03.2020
Lead beneficiary for the deliverable	University of Helsinki
Dissemination level of the deliverable	Public

Action coordinator's scientific representative
Prof. Mikko Kurimo
AALTO–KORKEAKOULUSÄÄTIÖ, Aalto University School of Electrical Engineering, Department of Signal Processing and Acoustics
mikko.kurimo@aalto.fi

MeMAD project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780069. This document has been produced by theMeMAD project. The content in this document represents the views of the authors, and the European Commission has no liability in respect of the content.

Authors in alphabetical order		
Name	Beneficiary	e-mail
Jorma Laaksonen	Aalto University	jorma.laaksonen@aalto.fi
Umut Sulubacak	University of Helsinki	umut.sulubacak@helsinki.fi
Jörg Tiedemann	University of Helsinki	jorg.tiedemann@helsinki.fi

Internal reviewers in alphabetical order		
Name	Beneficiary	e-mail
Sabine Braun	University of Surrey	s.braun@surrey.ac.uk
Tiina Lindh-Knuutila	LLS	tiina.lindh-knuutila@lingsoft.fi

Abstract

In this deliverable, we report on our final releases of machine translation models and tools that were developed through our efforts in WP4.

We introduce OPUS-MT, a WebSocket-based translation server, and our release of the MeMAD subtitle translation pipeline, both including pre-trained models suitable for general-purpose translation. Next, we introduce and discuss the `subalign` toolbox, and its key utilities that implement heuristics to convert between plain sentences and SRT-formatted subtitle segments with time codes. In connection with this, we introduce fine-tuned models for subtitle translation with the capability of token alignment for improved synchronisation. Afterwards, we introduce our releases of the MeMAD image caption translation and end-to-end speech translation systems. These systems are based on our work on multimodal machine translation discussed previously in D4.1, evaluated as part of our submissions to the WMT 2018 multimodal translation and IWSLT 2019 speech translation shared tasks, respectively. Furthermore, we also introduce our release of the MeMAD document-level translation models, which were developed through our experiments on discourse-aware machine translation, and evaluated as part of our submission to the WMT 2019 document-level translation shared task. Finally, we also describe our release of a dataset tailored for benchmarking document-level machine translation performance.

All of our software releases are open source with permissive licences of use, and our pre-trained models have been made freely available for download following the guidelines for open access. Our scripts and documentation have been organised into individual repositories located in the common MeMAD Github space, linking to the relevant pre-trained models (where applicable) hosted in the MeMAD community space on Zenodo. We include additional explanations and usage instructions in this deliverable as necessary.

Contents

1 Introduction

The purpose of this document is to collect information about released models, resources and tools in connection with the activities in WP4 on multilingual, multimodal and discourse-aware machine translation (MT) within the MeMAD project. In particular, the releases build on the work described in deliverables D4.1 (report on multimodal machine translation) and D4.2 (report on discourse-aware machine translation for audio-visual data). For convenience, we link the essential outcomes from a dedicated repository on MeMAD workpackages, which is available from GitHub.[1] Here, we will not repeat research and results described in various publications and deliverables that are connected with the models and their development but rather provide links to the released resources and brief information about their use. Nevertheless, we add findings and results that have not been reported before including benchmarks on project-internal test sets and multilingual translation models that have been prepared for this deliverable.

In particular, we include resources for the following categories:

- **General-purpose translation**: Our efforts in developing general-purpose MT models for the focus languages of the MeMAD project trained on large datasets.
- **Subtitle translation**: Tools for translating subtitles, including the extraction of appropriately segmented text, and the alignment of translations to given time frames.
- **Image caption translation**: Multimodal MT models optimised for the translation of image captions, as the outcome of our efforts in the WMT 2018 shared task.
- **Spoken language translation**: Multimodal MT models for the translation of speech to text in another language, connected to our submissions to IWSLT shared tasks.
- **Discourse-aware translation**: Document-level MT models that process an extended context to capture discourse-level dependencies across sentence boundaries.

In addition to the models that we release, we also publish our tools, scripts and pipelines to run these models, as well as information about their training and development. As long as copyrights permit, we also release datasets for further research and replicability.

The general principle for our releases is to provide our resource with permissive licences to enable their wide application. For software and tools we adopt MIT[2] and Apache[3] licences and for the models we focus on a permissive Creative Commons licence (CC BY 4.0).[4] Software tools and documentation are stored on our project space on Github[5] and the released models and other resources are published in our project community space at Zenodo[6].

In the following sections, we list our published material together with brief instructions.

[1] https://github.com/MeMAD-project/workpackages
[2] https://mit-license.org/
[3] https://apache.org/licenses/LICENSE-2.0
[4] https://creativecommons.org/licenses/by/4.0/
[5] https://github.com/MeMAD-project
[6] https://zenodo.org/communities/memad

2 General-purpose NMT

MeMAD focuses on six European languages: Dutch, English, Finnish, French, German and Swedish. Translation is a key component in the project, enabling cross-lingual access to audiovisual information in various workflows. Translation is required in intralingual subtitling, cross-lingual information enrichment, cross-lingual entity linking, and cross-lingual search. The MeMAD prototype implements a modular toolbox for the production use case, and incorporates many of those tasks. Furthermore, multilingual subtitle production outside of that prototype is another use case that we have studied intensively in the project.

Providing the functionality of a general-purpose tool for a translation component has been one of the major efforts in WP4. Here, we emphasise the practical and versatile use of such tools and the coverage of all language pairs included in the project setup. The releases include:

- Pre-trained models for all translation directions on large, mixed datasets with state-of-the-art neural MT.
- Translation server applications for deploying scalable services that can easily be integrated in different workflows.
- Scripts and pipelines for training these models.

2.1 Translation models

Released earlier by the University of Helsinki, the OPUS-MT[7] system (Tiedemann and Thottingal, 2020) includes a collection of publicly available general-purpose MT models. We have integrated the models corresponding to our language pairs of interest into the Limecraft Flow platform, where the majority of MeMAD use case and proof-of-concept evaluations were carried out, as general-purpose MT tools available on demand. In the course of these evaluations, these models have been used to facilitate metadata enrichment, cross-lingual search, and interlingual subtitling, among other things. OPUS-MT models are organised in an auxiliary repository hosted on Github[8], where download links can be retrieved.

We have also released a set of MT models as part of our subtitle translation system release,[9] which contains 30 text-based bilingual models (all possible translation directions among the six MeMAD languages). These models were intended to take part in the subtitle translation pipelines developed for the subtitling productivity evaluations, but follow the same training setting as the OPUS-MT models, and use similar training data. Therefore, the MT models that were enclosed in this release also constitute viable general-purpose MT models. These models[10] are available through the MeMAD community space on Zenodo, with open access under the Creative Commons Attribution CC BY 4.0 licence. For a detailed description of the full pipeline as well as other models tailored for subtitle translation, please refer to Section 3.

Both sets of our general-purpose MT models are based on the transformer (Vaswani et al., 2017) implementation of Marian (Junczys-Dowmunt et al., 2018), an open source neural MT framework. The transformers were set up with 6 encoder and 6 decoder layers, 8 attention

[7] https://github.com/MeMAD-project/Opus-MT
[8] https://github.com/MeMAD-project/Opus-MT-train/tree/master/models
[9] https://github.com/MeMAD-project/subtitle-translation
[10] https://zenodo.org/record/4389209 and https://zenodo.org/record/4556121

heads, and a dropout rate of 0.1. The training procedure uses an Adam optimiser with a learning rate of 0.0003, with linear warmup for the first 16 000 batches, and inverse square root decay through the remaining ones. Decoding for validation during training uses perplexity as the validation metric, and beam search with a beam size of 12. For replicability purposes, the exact training specification can be retrieved from the OPUS-MT models training script available from the corresponding repository on Github.[11]

(bleu↑)	! de	! en	!	! fr	! nl	! sv
de !	|	25:60	16:07	19:20	21:20	19:20
en !	29:16	|	22:90	27:11	29:86	28:56
!	14:08	16:96	|	14:16	16:57	17:78
fr !	21:39	25:28	16:76	|	20:85	19:26
nl !	22:27	27:22	18:79	20:62	|	22:20
sv !	21:21	26:88	21:36	20:21	23:70	|

Table 1: BLEU scores from benchmarking the MeMAD subtitle translation models on the held-out internal development sets sampled from the OpenSubtitles corpus. The scores were calculated each on a random selection of 100 000 sentence pairs, sampled from movies released in even-numbered years between 1970 and 2018 (inclusive), for which the OpenSubtitles 2018 release had data available in all 6 focus languages. Subtitles for movies released in odd-numbered years have been instead used as training data.

The OPUS-MT models were trained using all parallel data available for each corresponding language pair in OPUS (Tiedemann, 2012), a large collection of open parallel corpora for training MT models. Similarly, the training data for the subtitle translation models have been compiled from OPUS, except a small multi-parallel sampling of subtitle data from the OpenSubtitles[12] corpus was excluded from training, and instead held out as an internal development set (see Table 1 for the relevant performance metrics). Since the OPUS collection is always evolving, and the training sets for the two sets of models have been compiled about eight months apart, there should also be other minor differences resulting from OPUS corpora that were added or updated in this window. Furthermore, OPUS-MT contains one model per translation direction, trained until convergence with early stopping, reverting to the best snapshot of the model after 10 consecutive validation cycles in which the model did not improve. In contrast, the subtitle translation pipelines include ensembles of 5 randomly-seeded models for each direction, each model trained equally for 72 hours on 4 parallel Nvidia V100 GPUs.

OPUS-MT is meant to be used as a translation service, and was packaged to facilitate the process of setting up a server (see Section 2.2 for further details and instructions). In contrast, the subtitle translation models were intended to be set up locally, as modules for a pipeline to process SRT-formatted subtitle data. Nonetheless, an option to perform translation on plain text data (segmented as sentences, rather than subtitles) was built into the translation interface. This option can be activated via the `--plain-text-mode` flag, as in the invocation example below:

[11] https://github.com/MeMAD-project/OPUS-MT-train/blob/master/lib/train.mk
[12] https://www.opensubtitles.com

(bleu↑)	□ ! de	□ ! en	□ ! fr
de ! □	|	32.97	29.28
en ! □	28.41	|	|
fr ! □	24.31	|	|

Table 2: BLEU scores from benchmarking the MeMAD subtitle translation models on WMT 2020 test sets. Translation models other than de $ en and de $ fr did not have corresponding test sets in the 2020 release, and previous years of releases were not used for benchmarking due to a partial overlap with the training data.

```
./translate.py --src-lang de \
               --tgt-lang en \
               --input your/data/sample.de \
               --output your/data/sample.en \
               --gpu-devices 4 \
               --verbose \
               --log process.log \
               --plain-text-mode
```

Note that, while this option skips the initial sentence parsing and final subtitle segmentation steps, it still uses the other pipeline modules (i.e. preprocessing, restoration, and postprocessing). All the software dependencies listed in the `subtitle-translation` repository, except for `OpusTools-perl` and `subalign`, must still be installed and configured. Likewise, segmentation, restoration, and translation models must still be downloaded and unpacked as instructed. Tables 1 and 2 show benchmarking results for the subtitle translation models when used in this way, on the held-out development sets, and on the WMT 2020 test sets, respectively.

2.2 Software for building translation services

We have implemented a WebSocket translation service application based on Marian that can be deployed on modern GNU/Linux distributions using the setup published in OPUS-MT. The implementation includes a translation router daemon process that connects an arbitrary number of individual translation services that may contain multilingual as well as domain-specific services using a simple WebSocket API. A simple JSON configuration file can be used to specify the services to be included. Figure 1 shows an example for a configuration with two translation services, one for translating Finnish to English running on the same machine as the translation router (localhost), and another service for translating French to Estonian or Finnish running on a remote machine.

The API can be called using another simple JSON input format specifying the text to be translated, the source language code and the target language code:

```
{
    "localhost:20000" : {
        "source-languages" : "fi",
        "target-languages" : "en"
    },
    "192.168.1.14:21100" : {
        "source-languages" : "fr",
        "target-languages" : "et+fi"
    }
}
```

Figure 1: Translation router configuration.

```
{
    "text": "Mitä kuuluu?",
    "source": "fi",
    "target": "en"
}
```

The source language can also be omitted and the server will in that case try to automatically detect the input language using the Chrome compact language detection library (version 2). However, for short messages, this may not be very reliable. The developers mention that the software is designed for web pages of at least 200 characters and that it is not expected to do well very short text, lists of proper names, part numbers, etc. [13]. Therefore, please, use this option with care.

The system also supports domain-specific models assuming that several task-specific models have been deployed in the backend. The use of alternative models can be activated by adding an attribute "model" to the description of the translation system in order to separate it, for example, from other domains or other task-specified applications. Figure 2 shows and example for two different German-Finnish translation services with one of them optimised for the WMT news translation task.

The model can be specified in the API call by adding the model argument in the request:

```
{
    "text": "Wie geht's?",
    "source": "de",
    "target": "fi",
    "model": "wmt"
}
```

The result of API requests is also formatted in JSON providing various types of output (see Figure 3):

[13] https://github.com/CLD2Owners/cld2

```
{
    "192.168.1.19:20004" : {
        "source-languages" : "de",
        "target-languages" : "fi"
    },
    "192.168.1.12:20008" : {
        "model" : "wmt",
        "source-languages" : "de",
        "target-languages" : "fi"
    }
}
```

Figure 2: Translation router configuration with domain-specific models

- `result`: translation result in plain text
- `source-sentences`: list of source sentences segmented in plain text
- `target-sentences`: list of translated target sentences in plain text
- `source-segments`: list of source sentences segmented as subword units
- `target-segments`: list of translated target sentences segmented as subword units
- `alignment`: cross-lingual token alignments for each sentence (subword unit alignment)
- `source/target/server`: source language, target language, and translation server used

The segmented input and output are handy for the token alignments provided by the model. Note that the token alignment comes from cross-lingual attention, and that a model must be trained with the guided alignment feature of Marian in order to make this information useful for any further processing.

The example in Figure 3 illustrates BPE-segmented model output that has been pre-tokenised as well. For most of our current models, we skip tokenisation and run SentencePiece subword segmentation on raw text instead. All pre- and post-processing will be handled internally in the translation server and should not affect the format of the final result retrieved from the service except for attributes that provide the segmented strings from the internal representations.

2.3 Deploying translation servers

Installing the translation service software is straightforward and has been tested on Ubuntu Linux distributions versions 14.04, 16.04 and 18.04. Here are the main steps to be taken:

- Clone the software and install all pre-requisites

```
git clone https://github.com/Helsinki-NLP/Opus-MT.git
cd Opus-MT/install
```

```
f
    "alignment": [
        "0-0 0-2 1-1 2-3",
        "0-0 1-1 3-2 4-3 5-4"
    ],
    "result": "How are you? The translation is fun.",
    "server": "192.168.1.18:20001",
    "source": "fi",
    "source-segments": [
        "Mit\u00e4 kuuluu ?",
        "K\u00e4\u00e4@@ nn\u00f6@@ s on hauskaa ."
    ],
    "source-sentences": [
        "Mit\u00e4 kuuluu?",
        "K\u00e4\u00e4nn\u00f6s on hauskaa."
    ],
    "target": "en",
    "target-segments": [
        "How are you ?",
        "The translation is fun ."
    ],
    "target-sentences": [
        "How are you?",
        "The translation is fun."
    ]
g
```

Figure 3: Translation output from the WebSocket server. Non-ASCII characters are encoded with their corresponding Unicode character code with the JSON speci c encoding scheme. Source and target segments are tokenized in this example and source segments are also segmented into subword units using BPE. The token separator is '@@' indicating that the subsequent space character is to be deleted.

```
make all
sudo make install
cd ..
```

- Adjust the configuration file to meet your plans about the server to be run
- Download and set up a specific language pair (make sure that a model exists for that language pair), here we show the example of Finnish to English:[14]

```
sudo make SRC=fi TRG=en OPUSMT_PORT=10000 MARIAN_PORT-20000 all
```

[14] Admin rights are necessary to install the software daemon and startup scripts in default locations. The models and translation cache database is also set up in globally shared directories that require admin rights to modify in standard Linux systems.

This should start 3 daemons that run the service:

- The Marian NMT server that translates plain text segmented into subword units (running on `MARIAN_PORT`)
- The language-specific OPUS-MT server (running on `OPUSMT_PORT`) that performs pre- and post-processing and interacts with the Marian NMT server
- the OPUS-MT router server that connects various individual OPUS-MT services (running on port 8080 by default)

Configuration files and models will be installed in `/usr/local/share/opusMT/` and startup scripts for the individual services are in `/etc/init.d`.

Starting an additional translation service is easy by re-running the installation script with new parameters, for example a service for translation German to English:

```
sudo make SRC=de TRG=en MARIAN_PORT=10001 OPUSMT_PORT=20001 opusMT-
    server
```

It is still necessary to edit the configuration file to include the new server; add

```
    "localhost:20001" : {
        "source-languages" : "de",
        "target-languages" : "en"
    },
```

and re-install the configuration, and re-start translation services and the router daemon:

```
sudo make opusMT-router
sudo service marian-opus-de-en restart
sudo service opusMT-opus-de-en restart
sudo service opusMT restart
```

Note that the translation servers also build up a cache for efficiency reasons to avoid re-translating identical sentences that have been translated by the same model before. The cache is stored using an SQLite database in `/var/cache/opusMT/`.

Additional recipes for removing services and further details are given in the documentation and the makefile in the OPUS-MT repository[15].

3 Subtitle translation

This section describes tools and resources tailored towards the translation of subtitles. We present pre- and post-processing tools and models that are optimised for subtitle translation.

[15] https://github.com/Helsinki-NLP/Opus-MT/blob/master/Makefile

3.1 Pre- and post-processing

Subtitle translation requires some special treatment to be used with standard machine translation models. The `subalign` software package implements various tools to perform the necessary pre- and post-processing to work with SRT files and their translation. This enables a streamlined pipeline of subtitle translation with regular, sentence-based translation engines. For document-level approaches, please refer to Section 6.

The toolbox includes the following scripts:

`srt2xml`: A tool that converts subtitles in SRT format to simple OPUS-style XML format. It performs sentence splitting and tokenisation using regular expressions and language-specific non-breaking prefixes (taken from the Europarl corpus version tools). The tool also converts between character encodings (using explicit BOM detection or explicit parameters), and implements various heuristics to merge lines in cases of sentences that continue on subsequent lines as well as in subsequent subtitle blocks. It also splits subtitle blocks in case of detected sentence boundaries, and produces sentence boundary markup while keeping time information in place.

`srtalign`: A tool for aligning subtitle files based on time information. The system looks for sentence alignments that maximise the time overlap based on the output produces by `srt2xml`. Time information is extrapolated to match sentence boundaries based on a simple linear correlation between the length of characters and the time span dedicated to that string. Furthermore, the tool implements synchronisation procedures (Tiedemann, 2008) based on lexical anchor points that can be detected using cognate heuristics or bilingual dictionaries. The `subalign` toolbox provides 361 dictionaries extracted from automatic word alignment (about 360,000 dictionary entries altogether).

`mt2srt`: A tool that aligns translated text to a given subtitle template to fill the given time slots with information coming from an MT system. The tool implements a length-based approach for this alignment, using various adjustments for the subtitle alignment case. Basically, it restricts the alignment to 1-to-*m* alignment types, using the original text in a given time slot as the source segment, and sentence fragments from the translations as the segments in the target language. For this, translated sentences are split into clauses based on punctuation characters, and the procedure finds the globally optimal alignment that minimises the costs based on the length correlation factor. Additional heuristics can be used to constrain the maximum length of a subtitle block and to penalise subtitle breaks within a running sentence. The template can be either in XML or in SRT format, and the input should be plain text with one sentence per line. All data should be encoded in UTF-8. More details on subtitle block alignment are provided in Koponen et al. (2020b).

The MeMAD subtitle translation pipeline that makes use of these tools was released through the `subtitle-translation` repository[16] in the MeMAD Github space. We provide further details on the structure of the pipeline in the repository's documentation, as well as instructions for using the pipeline with pre-trained models, as discussed earlier in Section 2.1. In addition to the automatic reference-based metrics covered in this deliverable, we report our findings from further use case based evaluations of subtitling productivity and end user reception of translated subtitles in deliverable D6.9.

[16] https://github.com/MeMAD-project/subtitle-translation

3.2 Subtitle-optimised translation models

Subtitle translation requires further adjustments that impacts translation models as well. General-purpose models do not necessarily capture the specific properties and language style even if the training data contains large portions of in-domain training data. Furthermore, the alignment of translation to the audiovisual content requires further information to be provided by the translation engine. Therefore, we also produce models that are optimised for the use of this specific task using the following two steps:

Cross-lingual alignment: We train translation models that incorporate unsupervised word alignment into the training procedure in order to guide one of the cross-lingual attention heads to follow the links between tokens in source and target language.

Fine-tuning: We fine-tune models for the subtitle translation domain using a second step of additional training after creating a general-purpose translation model.

For the guided alignment feature, we use the efficient statistical word aligner `eflomal`,[17] which has been developed in the Language Technology research group in the University of Helsinki. We apply the software on the parallel training data segmented on the subword level using SentencePiece, in order to produce links that can directly be used by the neural MT model as a correspondence to cross-lingual token-level attention. `eflomal` is run on equally-sized partitions of the data, each with 5 million sentence pairs, to enable efficient and reliable alignments even on the big datasets that we are working with. We run word alignment for each language pair in both directions, and symmetrise the two directions using *grow-diag-final* heuristics (Koehn, 2009), a common strategy that emphasises recall and data coverage.

Besides bilingual translation models, we also train multilingual models in order to provide compact models that cover all the focus languages of the project. Multilingual models offer the possibility to deploy a single translation service that can be used for multiple translation directions and, hence, decrease resource requirements when running extensive services. Related work also reports positive effects through transfer learning when training multilingual models. However, in our case of high-resource languages we cannot see that effect and rather see slight drops in performance when using multilingual settings instead of strong bilingual models. Nevertheless, the remaining advantage of larger language coverage in a single model can still outweigh the minor decrease in translation quality depending on the task and its performance requirements.

The training set for the multilingual models is balanced between the individual language pairs, using a simple sampling strategy on shuffled data. In our case, we use one million sentence pairs per language pair to create a sufficiently large but still manageable dataset to train the system. Target language tokens are added to the system to enable the translation into various languages in the otherwise completely shared model among all translation directions. Language labels are given as ISO 639-3 codes enclosed between double inequality signs to distinguish them from regular words. For example, Swedish for the target language is encoded by adding a token >>*swe*<< to the beginning of the input string.

In summary, we include the following models:

[17] https://github.com/robertostling/eflomal

- Bilingual models: All combinations and translation directions for the six MeMAD focus languages, using all data available from the Tatoeba-MT challenge release.[18]
- A many-to-English translation model from all focus languages to English (without language labels), and sampled training data from the Tatoeba-MT challenge. Translations into English is a common use case and, therefore, we include this specific setup in our list of supported models.
- An English-to-many translation model that translates from English to any of the focus languages using unique language labels (same sampled data as above). Similar to above, translations from English are common in real-world use cases and, hence, this particular multilingual setup is important.
- A many-to-many translation model that covers all focus languages in all directions trained on data sampled from the same source as the other models. In contrast to the models above, this instance covers all translation directions between the focus languages of the MeMAD project.

For fine-tuning, we use the domain labels given by the Tatoeba-MT challenge release, and extract a sample of the subtitle section of each of the datasets (one million sentence pairs). Furthermore, we fine-tune on MeMAD internal data coming from Yle to contrast the models with the performance of clearly in-domain tuning. For the latter we reserved 10,000 sentences each for validation and kept the rest for training the fine-tuned models. The final models are then tested with an independent benchmark test set also provided by Yle with no overlap with the training and development data. This data set contains 1,254 sentence pairs for Finnish-Swedish general-purpose subtitles (FIN-SWE), 2,837 sentences that translate Finnish subtitles to Swedish subtitles for the hearing impaired (FIN-SWH) and 625 sentences that translate Finnish subtitles for the hearing impaired to general purpose subtitles in Swedish (FIH-SWE).

tune / test	FIN-SWE	FIH-SWE	FIN-SWH	SWE-FIN	SWH-FIN	SWE-FIH
baseline	22.3	17.0	18.2	20.8	15.9	12.2
OpenSubtitles (1M)	22.0	16.8	17.9	20.9	15.7	12.5
Yle-all (2M)	24.7	19.6	19.5	22.7	17.4	13.6
Yle-FIN-SWE (1.1M)	24.9	18.9	19.5	23.1	17.3	13.9
Yle-FIH-SWE (47k)	23.6	19.7	18.4	21.5	16.0	14.8
Yle-FIN-SWH (850k)	23.8	18.5	19.5	23.0	17.7	13.9

Table 3: Fine-tuning general-purpose NMT models (*baseline*) for the subtitle domain. Test sets include translations between subtitles for Finnish (FIN) and Swedish (SWE) with variants of subtitles for the hearing impaired (FIH and SWH, respectively). The rows refer to different tuning sets including one million parallel sentences sampled from OpenSubtitles (OPUS) and MeMAD internal training data provided by Yle with subsets for the specific language variants (general subtitles or for the hearing impaired).

Table 3 shows the results when evaluating on the internal YLE benchmarks. They demonstrate the importance of fine-tuning and the appropriateness of the data used for that purpose. The metrics show that generic subtitle data (from OpenSubtitles) is not good enough for domain adaptation as this material represents a wide variety of genres mostly (probably even exclusively) with a source language other than Finnish or Swedish. The performance actually deteriorates slightly when continuously optimising the pre-trained model for that data set. On

[18] https://github.com/MeMAD-project/Tatoeba-Challenge

the other hand, in-domain training data leads to clear improvements with a significant impact on the style match that refers to subtitles for the hearing impaired or the general audience. This is even more remarkable considering the small data set that we have available for the translation from and to Finnish for the hearing impaired showing the importance of examples that teach the model to take care of the inherent style difference.

We release all our models using our project community space on Zenodo[19] with links from the `subtitle-translation` repository[20] on Github. Unfortunately, the in-domain training and test data cannot be released together with the models due to copyright issues and license agreements. We are currently negotiating the release of the test set in order to provide a benchmark for comparing and replicating our results.

4 Image caption translation

The best-performing image caption translation model implemented within MeMAD was released through our Zenodo community[21], and the documentation is available from the `image-caption-translation` repository[22] on Github. The system constitutes our submission to the shared task at WMT 2018, and implements the winning system in that competition (Barrault et al., 2018). The system is carefully described in Grönroos et al. (2018) and we will not repeat the details here. This report focuses on the release details and provides links and installation instructions in order to deploy and run the model. Below we list the essential steps for setting up the system and refer the reader to the original publication to understand the architecture of the model.

Downloading the model

Fetch the model from Zenodo using the following command

```
curl https://zenodo.org/record/4038444/files/opennmt.transformer.
   multiling.mscoco%2Bmulti30k%2Bsubs3M.domainprefix.mmod.imgw.meanfeat
   .detectron.mask_surface.bpe50k_acc_80.57_ppl_2.43_e23.pt?download=1
   --output models/opennmt.transformer.multiling.mscoco+multi30k+subs3M
   .domainprefix.mmod.imgw.meanfeat.detectron.mask_surface.
   bpe50k_acc_80.57_ppl_2.43_e23.pt
```

Installing the software

Start by cloning the github repository at `https://github.com/MeMAD-project/image-caption-translation.git`.

We recommend a conda-based installation and provide the corresponding commands for installing the codebase below. The software used CUDA by default but can also run on CPU.

[19] https://zenodo.org/record/4556121
[20] https://github.com/MeMAD-project/subtitle-translation
[21] https://zenodo.org/record/4038444
[22] https://github.com/MeMAD-project/image-caption-translation

Feature extraction software can be installed in the following way:

```
conda create --name memaddetectron2 --file env/detectron2.cuda.conda -
   c pytorch
source activate memaddetectron2
pip install -r env/detectron2.cuda.pip
source deactivate
```

The installation of the translation system is described below. Note that this codebase uses specific versions of some libraries, which is specified in the environment files included in the repository used by the installation commands below. CUDA is disabled.

```
conda create --name memadmmt --file env/mmt.nocuda.conda -c pytorch
source activate memadmmt
pip install -r env/mmt.nocuda.pip
git clone https://github.com/Waino/OpenNMT-py.git
pushd OpenNMT-py
git checkout develop_mmod
python setup.py install
popd
source deactivate
```

Using the model

First of all, we need to extract `detectron2` features and store them in `img_feat.npy`:

```
source activate memaddetectron2
tools/image-features.py --imglist data/imglist
source deactivate
```

Note the following settings: `img_feat_dim=80`, `dtype=torch.float32`, saved as an $(N; 80)$ matrix in NumPy `.npy` format, where N is the number of lines to translate.

The next step is to apply BPE segmentation to previously tokenised and lowercased text:

```
source activate memadmmt
tools/apply_bpe.py --codes models/bpe.50k.multiling < data/input >
   data/segmented
```

After that, prepend the target language tag (either `TO_de` or `TO_fr`) and the domain tag:

```
sed -e "s/^/<TO_de> <DOMAIN_caption> /" < data/segmented > data/prefixed.de
sed -e "s/^/<TO_fr> <DOMAIN_caption> /" < data/segmented > data/prefixed.fr
```

Finally, we are able to perform translations:

```
OpenNMT-py/translate_mmod_finetune.py \
    -model models/opennmt.transformer.multiling.mscoco+multi30k+subs3M.domainprefix.mmod.imgw.meanfeat.detectron.mask_surface.bpe50k_acc_80.57_ppl_2.43_e23.pt \
    -src data/prefixed.de \
    -path_to_test_img_feats img_feat.npy \
    -output data/translated.de \
    --multimodal_model_type imgw
```

The translations still need to be post-processed in order to join BPE subwords and to recase the output.

As described in the paper (Grönroos et al., 2018), it is also possible to feed zero vectors as dummy features by replacing `-path_to_test_img_feats img_feat.npy` with `-path_to_test_img_feats dummy.zeros.npy` in the above command.

5 Spoken language translation

Whenever we needed to translate spoken language in MeMAD, we opted for a cascaded approach. This approach realises translation from audio containing spoken language by pipelining (1) automatic speech recognition (ASR), and (2) text-based machine translation (MT) stages. Essentially, this pipeline breaks the task of multimodal translation down into modality conversion followed by unimodal translation.

Our experiments on subtitling productivity and the user reception of automatically-generated subtitles, as part of the Use Case UC4 evaluations, both featured such pipelines. However, these have been cross-work-package efforts, integrated together under WP6. While the MT components were developed in the University of Helsinki and released within WP4, the ASR components were provided by Lingsoft and Aalto University as part of WP2. The scope of this deliverable only includes the MT components, however, the full speech translation pipeline can be reproduced by first generating a transcript of the audio using Lingsoft ASR, and continuing with the MT pipeline from the sentence segmentation step. We report further details of the pipeline as utilised for subtitle translation in deliverables D6.6 and D6.9.

We have also experimented with end-to-end speech translation, where a monolithic system undertakes the translation of source language audio to target language text in a single stage. Our release includes two sets of translation models that are able to translate between English and German in either direction—one which only processes source language audio, and an-

other which can also make use of transcripts of the audio as auxiliary text input. Although the system performs multimodal translation end-to-end, the translation process still involves offline preprocessing and postprocessing steps. The preprocessing step is required when translating with transcripts, and performs normalisation, truecasing, and subword segmentation on the text input. The postprocessing step converts the translation output to a human-readable plain text format, and is required for both audio-to-text and audio+text-to-text translations.

Our end-to-end speech translation models were developed, along with the subtitle translation pipelines, in preparation for the MeMAD submissions to the IWSLT offline speech translation task in 2020 (Vázquez et al., 2020). Unlike the subtitle translation pipelines, the end-to-end speech translation models remained unchanged since our 2020 submission. The final version of the system has been released via the `speech-translation` repository[23] in the MeMAD Github space, including download links for the pre-trained models archived on Zenodo, and detailed instructions for setting up the system.

6 Discourse-aware translation

This deliverable includes two released packages in relation to discourse-aware machine translation: (1) Pre-trained concatenation models for three language pairs and (2) a dataset for benchmarking document-level approaches to machine translation.

6.1 Document-level models

We release six pre-trained models for concatenation-based document-level translation. In particular, we provide models that translate between Finnish and three other languages, English, French and Swedish in both directions for each language pair. The models can be downloaded from https://doi.org/10.5281/zenodo.4287562.

The models were trained using Marian (v1.8.2), implementing state-of-the-art transformer architectures, using 6 layers in both the encoder and the decoder, with 8 self-attention heads per layer. We use SentencePiece for tokenisation and subword segmentation and train each model on large collections of human translations provided by OPUS, the open parallel corpus.

The document-level models apply a so-called concatenation approach in which larger chunks are concatenated in order to enable cross-sentence dependencies to be covered by the model. In our case, we use text windows of up to 100 tokens on both sides and apply a simple greedy segmentation approach to create those chunks. Table 4 summarises the size of each data collection we use for training. We can see, that we have substantial amounts of data in each language pair ranging from 30 million to 44 million sentence pairs that constitute between 5.7 and 8.5 million document-level segment pairs to train on. This means that the window of 100 tokens creates segments with 5–6 sentences on average.

We also provide the scripts to pre- and post-process data that need to be translated with those models, and the procedures are straightforward (assuming a specific benchmark dataset and model in this case):

[23] https://github.com/MeMAD-project/speech-translation

language pair	segment pairs	sentence pairs
! sv	5,712,031	30,604,442
! en	8,494,683	43,942,504
! fr	6,087,869	30,138,134

Table 4: Sizes of the training data used for document-level MT models.

```
1 spm_encode --model models/fi-en/opus.src.spm32k-model < data/
    newstest2019-fien-src.fi.txt | scripts/split-text.pl -l 100 > data/
    data/newstest2019-fien-src.fi.doc100
2 marian-decoder -c models/fi-en/decoder.yml < data/newstest2019-fien-
    src.fi.doc100 > data/newstest2019-fien-sys.en.doc100
3 scripts/post-process.sh < data/newstest2019-fien-sys.en.doc100 > data/
    newstest2019-fien-sys.en.txt
```

The commands above perform the essential pre-processing steps (line 1) including subword segmentation using a pre-trained SentencePiece model provided in the release followed by a script that splits the text into chunks of maximum 100 tokens. The size needs to match the model that is applied in the subsequent step, which calls the NMT decoder (line 2) with the input created in the first step. The final step performs simple post-processing steps mainly referring to merging subword units produced by the NMT decoder. The prerequisites for running the translation with the steps above are a successfully compiled Marian decoder and the SentencePiece subword segmentation software installed in the path of your your system.

A detailed analysis of document-level models and their impact on the work of professional translators is provided in Koponen et al. (2020a).

6.2 Benchmarks

We release the datasets used in our study on concatenation approaches to document-level machine translation published at DiscoMT 2019 (Scherrer et al., 2019). They are taken from the English–German news translation task at WMT 2019 and the English–German bitext in the OpenSubtitles 2016 collection from OPUS. All datasets are sentence-aligned with corresponding lines being aligned to each other. Document boundaries are marked with empty lines (on both sides of the parallel corpus). The release contains a dedicated split into development, test and training data in the two domains considered in the study. The package can be downloaded from https://zenodo.org/record/3525366.

The following list shows the essential content of the release along with individual file sizes:

```
1     dev
2         ost.tok.de              6.2 MB
3         ost.tok.en              5.8 MB
4         wmt.tok.de            663.1 kB
5         wmt.tok.en            592.5 kB
```

```
    test
        ost.tok.de              188.2 kB
        ost.tok.en              203.3 kB
        ost.nocontext.tok.de    193.2 kB
        ost.nocontext.tok.en    208.3 kB
        ost.shuffled.tok.de     188.3 kB
        ost.shuffled.tok.en     203.3 kB
        wmt.tok.de              397.1 kB
        wmt.tok.en              360.6 kB
        wmt.nocontext.tok.de    400.0 kB
        wmt.nocontext.tok.en    363.5 kB
        wmt.shuffled.tok.de     397.1 kB
        wmt.shuffled.tok.en     360.6 kB
    train
        ost.tok.de              516.4 MB
        ost.tok.en              479.7 MB
        wmt.de-en.tok.de        504.1 MB
        wmt.de-en.tok.en        435.2 MB
        wmt.en-de.tok.de          1.7 GB
        wmt.en-de.tok.en          1.5 GB
    unshuffle.py
```

The interesting part of the benchmark is the inclusion of shuffled and non-contextualised variants of the test data that makes it possible to evaluate systems with respect to their use of discourse-level dependencies. The assumption is that shuffled context or no context at all should show up in the evaluation scores hurting systems that rely on discourse-level features as part of their decision process.

7 Conclusion

This deliverable has presented all of our releases of tools and models developed in connection with the main tasks of MeMAD WP4. Our general-purpose machine translation models and spoken language translation systems have been widely-used catalysts for project-wide collaboration, and likely the most valuable technology among the contributions of WP4 in the larger scheme of MeMAD. As a consequence, our corresponding model and software releases make up a large portion of our public releases. Our efforts in developing translation systems that involve multimodality (image caption translation, spoken language translation) and discourse-awareness (document-level machine translation) have demonstrated that they serve relatively small niches. Our initial work on these systems has been in accordance each with a corresponding public shared task, so that we would become familiar with the state of the art, and avail ourselves of training and test datasets, as well as targeted evaluation methods. While our image caption translation and document-level machine translation releases correspond to the outcomes from our tasks for 2018 (as described in D4.1) and 2019 (as described in D4.2) respectively, we have expanded on both in the final year of MeMAD. In regard to multimodality, we have finalised our pipeline formula for semi- or fully-automatic interlingual subtitling of media containing spoken language. In the context of discourse-aware machine translation, we

have introduced a benchmarking set tailored for document-level machine translation in order to facilitate further research, as the culmination of our work on the subject.

All WP4 releases have been made open for public use under permissive licences. In general, we have made our model releases as open access deposits via the MeMAD Zenodo community. Each deposit includes a summary description, and links to further resources and documentation as required. Our software releases have been made through code repositories registered under the MeMAD Github organisation. Each such repository contains the full source code as well as documentation that lists software dependencies, installation/usage instructions and examples, and/or external links to pre-trained models and datasets as necessary. We trust that our releases of tools and models will make life easier for others wishing to reproduce our results, use our outputs, and hopefully build upon them to advance the state of the art.

References

Loïc Barrault, Fethi Bougares, Lucia Specia, Chiraag Lala, Desmond Elliott, and Stella Frank. 2018. Findings of the third shared task on multimodal machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 304–323, Belgium, Brussels. Association for Computational Linguistics.

Stig-Arne Grönroos, Benoit Huet, Mikko Kurimo, Jorma Laaksonen, Bernard Merialdo, Phu Pham, Mats Sjöberg, Umut Sulubacak, Jörg Tiedemann, Raphael Troncy, and Raúl Vázquez. 2018. The MeMAD submission to the WMT18 multimodal translation task. In *Proceedings of the Third Conference on Machine Translation*. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020a. Mt for subtitling: Investigating professional translators' user experience and feedback. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA), 1st Workshop on Post-Editing in Modern-Day Translation*, pages 79–92.

Maarit Koponen, Umut Sulubacak, Kaisa Vitikainen, and Jörg Tiedemann. 2020b. MT for subtitling: User evaluation of post-editing productivity. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 115–124, Lisboa, Portugal. European Association for Machine Translation.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level NMT in two different domains. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, Hong-Kong. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT – building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 479–480, Lisboa, Portugal. European Association for Machine Translation.

Jörg Tiedemann. 2008. Synchronizing translated movie subtitles. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). Http://www.lrec-conf.org/proceedings/lrec2008/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Raúl Vázquez, Mikko Aulamo, Umut Sulubacak, and Jörg Tiedemann. 2020. The University of Helsinki submission to the IWSLT2020 offline SpeechTranslation task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 95–102, Online. Association for Computational Linguistics.

A Appendices

Below we attach a number of recent papers that demonstrate the use of our models and tools in professional workflows (papers A.1 and A.2) and our efforts related to the development of speech-to-text translation systems and resources for open machine translation development (papers A.3, A.4 and A.5).

Appendix A.1: A paper that discusses the experience and feedback of professional translators when working with MT integrated in the workflow of subtitle translation. Published at AMTA 2020

Appendix A.2: An evaluation of post-editing productivity in subtitling with machine translation. Published EAMT 2020.

Appendix A.3: A short paper on the development of open NMT models and tools that cover a large range of languages. Published at EAMT 2020.

Appendix A.4: A system paper describing the MeMAD submission to the offline speech translation task at IWSLT 2020.

Appendix A.5: A research paper presenting a new large scale resource and benchmark for machine translation with a large coverage of languages and language combinations. Published at WMT 2020.

MT for Subtitling: Investigating professional translators' user experience and feedback

Maarit Koponen maarit.koponen@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Umut Sulubacak umut.sulubacak@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Kaisa Vitikainen kaisa.vitikainen@yle.fi
Yleisradio Oy, Helsinki, Finland

Jörg Tiedemann jorg.tiedemann@helsinki.fi
Department of Digital Humanities, HELDIG, University of Helsinki, Helsinki, Finland

Abstract

This paper presents a study of machine translation and post-editing in the field of audiovisual translation. We analyse user experience data collected from post-editing tasks completed by twelve translators in four language pairs. We also present feedback provided by the translators in semi-structured interviews. The results of the user experience survey and thematic analysis of interviews shows that the translators' impression of post-editing subtitles was on average neutral to somewhat negative, with segmentation and timing of subtitles identified as a key factor. Finally, we discuss the implications of the issues arising from the user experience survey and interviews for the future development of automatic subtitle translation.

1 Introduction

Developments in translation technology and machine translation (MT), particularly the quality improvements achieved by neural machine translation (NMT) in recent years, have led to MT increasingly becoming part of the modern-day translators' toolkit. Although post-editing (PE), where MT is used to produce a raw translation output which is then checked and corrected by a translator, has increased in many areas of translation, its use remains uncommon in audiovisual translation (AVT). AVT approaches include dubbing, voice-overs and subtitling for the purpose of making AV content accessible to audiences with no or limited understanding of the language of the original content. Different approaches are used to varying degrees depending on the type of content (e.g. voice-overs are common for documentaries) and region (e.g. subtitling is the predominant practice in Northern European countries).

As studies and practical experience have shown potential for PE in increasing productivity in other forms of translation, interest in implementing MT tools and PE workflows has also grown in the AV field. Studies have explored the use of MTPE subtitle translation with some promising although mixed results regarding effect on productivity (e.g. Bywood et al., 2017). When exploring the usability of such tools, however, productivity measurement is only one aspect. As Etchegoyhen et al. (2014) argue, subjective feedback from translators is equally important, as it provides insight into the actual user experience and necessary improvements.

This paper presents a pilot study investigating the usability of MT and PE in the subtitling

Proceedings of the 14th Conference of the Association for Machine Translation in the Americas
October 6 - 9, 2020, 1st Workshop on Post-Editing in Modern-Day Translation

workflow from the perspective of the prospective users. Twelve professional subtitle translators working in four language pairs (Finnish↔Swedish and Finnish↔English) subtitled short video clips by post-editing MT output. We analyse feedback collected with a user experience questionnaire and semi-structured interviews for positive and negative evaluations of the PE experience and improvement suggestions. We start with an overview of related work on MT and PE in the subtitling context and work on user feedback (Section 2). After describing our approach to automatic subtitle translation (Section 3), and the subtitle PE experiment (Section 4), we present the questionnaire and interview analyses (Section 5), followed by discussion of the observations and our ongoing work based on these analyses.

2 Related work

2.1 Subtitling, MT and PE

Subtitle translation differs from translating purely textual material in that the source text consists of the spoken audio, together with the visual mode, while the target text is a written representation of translated speech. Due to technical limitations like the number of characters within a subtitle frame and the time each subtitle remains visible, paraphrasing and condensation are typical features of subtitle translation (see e.g. Pedersen, 2017). The work of subtitle translators may involve "first translation", where they translate from the source audio and determine the segmentation and timing of the subtitle frames ("spotting"), or translation with subtitle templates, where the source text consists of pre-existing intralingual subtitles in the source language or sometimes interlingual subtitles in a pivot language (often English) with set subtitle segmentation and timing (Nikolić, 2015).

To date, the use of MT and PE for subtitling has been less common in AVT than other translation fields. Explanations for this may include the characteristics of subtitle translation, which pose challenges for MT, and also the difficulty of integrating current NMT systems to subtitle translation workflows (Matusov et al., 2019). MT for movie and TV subtitling has been tested in some language pairs since the early 2000s (Melero et al., 2006; Volk et al., 2010; de Sousa et al., 2011) with suggestions that PE may increase productivity also in this context.

A subtitle-oriented statistical MT system and PE platform was developed by the SUMAT project, and tested in a user evaluation involving several language pairs and 19 professional subtitle translators (Etchegoyhen et al., 2014; Bywood et al., 2017). In a study comparing task time for translation from scratch and MTPE, Bywood et al. (2017) report that MTPE increased the translators productivity; however, the results varied for different translators, language pairs and content types. More recently, Matusov et al. (2019) tested an NMT system customised for subtitles using parallel subtitle corpora from OpenSubtitles, GlobalVoices and TED talks and reported productivity increases for MTPE in a study involving two translators.

So far, work has focused on the use of intralingual subtitles as the source text for MT, but a recent paper by Karakanta et al. (2020a) explores an end-to-end spoken language translation system for subtitling. No user evaluation of the system is reported, although Karakanta et al. (2020a) note that based on automatic evaluation against "gold standard" human subtitles the MT quality appears satisfactory. Karakanta et al. (2020b) also investigate annotating subtitle corpora for segment breaks and propose an approach for segmenting sentences into subtitles conforming to length constraints.

2.2 Studies on user experience/feedback from translators

Subjective feedback is invaluable for providing insight into tools and workflows that affect the actual work of the prospective users, and revealing issues that would not be evident from the translations or process data (see Bundgaard, 2017). Various studies have investigated professional translators' experience with and perceptions of MT and PE with questionnaires and

interviews. Analyses have reported mixed experiences: while translators sometimes find MT helpful, for example by providing useful terminology and making their work faster, other times PE may be even slower than translation from scratch. Whether working with technical or literary texts, translators often express concerns about MT affecting the final translation quality as well as their (cognitive) processes because the output can potentially mislead the translator or limit their creativity (e.g. Guerberof Arenas, 2013; Bundgaard, 2017; Moorkens et al., 2018).

Translator feedback on MT and PE in the context of AV translation was collected and analysed in the user evaluations of the SUMAT project (Etchegoyhen et al., 2014; Bywood et al., 2017). Etchegoyhen et al. (2014) describe a questionnaire used in the second evaluation round, where 19 translators carried out PE tasks in several language pairs and rated their impression of the PE process rather negatively overall (average 2.37 on a 5-point scale). Based on translator feedback, Etchegoyhen et al. (2014) identified improving MT quality to reduce cognitive load, improving quality estimation and filtering MT segments, and improving user interfaces for PE of MT subtitles as key issues for increasing usability.

Matusov et al. (2019) report a user experiment with two translators who both subtitled two programmes (a documentary and a sitcom) partly from scratch and partly with two different MT outputs. The translators rated their impression of the PE experience on average "fair" (3 on a 5-point scale) for the subtitle optimised system. The translator feedback noted useful terminology as one of the main reasons they would consider using MT in their work, but also expressed concerns about incorrect or unusual translations in the MT affecting the quality of the final translation (Matusov et al., 2019).

The study reported in this paper builds upon these analyses by collecting feedback on MT and PE for subtitling from professional subtitle translators. We aim to investigate the translators' impressions of PE more closely by introducing a more detailed user experience questionnaire where they rate different aspects of the process (see Section 4.3).

3 Automatic subtitle translation

Machine translation for subtitles requires some special treatment that we will discuss in this section. In particular, we consider models with extended context, which we will call *document-level translation models* and special tools that align translations with subtitle frames to be shown on screen. First, we briefly present the datasets and models before discussing frame alignment as a post-processing step.

3.1 Datasets and MT models

Our MT models are trained on a mix of diverse data sets[1] taken from OPUS.[2] Altogether, this includes over 30 million translation units for Finnish ↔ Swedish and about 44 million units for Finnish ↔ English. We follow the common practice in MT development to include as much data as possible even when coming from very different domains. However, the largest proportion of the training examples comes from a large collection of movie and TV show subtitles (the OpenSubtitles v2018 dataset) constituting almost half of the Finnish ↔ Swedish data and over 65% of the Finnish ↔ English data. This is certainly an advantage for our task and, hence, we expect a rather good domain-fit of our models.

We train both sentence-level and document-level models based on the Transformer architecture (Vaswani et al., 2017), the current state of the art in NMT. In particular, we apply the implementation from the MarianNMT toolkit (Junczys-Dowmunt et al., 2018), a production-ready software with fast training and decoding tools. The architecture refers to a 6-layered

[1] OPUS corpora used: bible-uedin, DGT, EMEA, EUbookshop, EUconst, Europarl, Finlex, fiskmo, GNOME, infopankki, JRC-Acquis, KDE4, MultiParaCrawl, OpenSubtitles, PHP, QED, Tatoeba, TildeMODEL, Ubuntu, wikimedia

[2] `http://opus.nlpl.eu`

A.2 EAMT paper on post-editing of subtitle MT

A.3 EAMT paper on OPUS-MT

A.4 MeMAD submission to the IWSLT shared task

A.5 WMT paper about the Tatoeba MT Challenge

The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT

Jörg Tiedemann
University of Helsinki
jorg.tiedemann@helsinki.fi
https://github.com/Helsinki-NLP/Tatoeba-Challenge

Abstract

This paper describes the development of a new benchmark for machine translation that provides training and test data for thousands of language pairs covering over 500 languages and tools for creating state-of-the-art translation models from that collection. The main goal is to trigger the development of open translation tools and models with a much broader coverage of the World's languages. Using the package it is possible to work on realistic low-resource scenarios avoiding artificially reduced setups that are common when demonstrating zero-shot or few-shot learning. For the first time, this package provides a comprehensive collection of diverse data sets in hundreds of languages with systematic language and script annotation and data splits to extend the narrow coverage of existing benchmarks. Together with the data release, we also provide a growing number of pre-trained baseline models for individual language pairs and selected language groups.

1 Introduction

The Tatoeba translation challenge includes shuffled training data taken from OPUS[1], an open collection of parallel corpora (Tiedemann, 2012), and test data from Tatoeba[2], a crowd-sourced collection of user-provided translations in a large number of languages. All data sets are labeled with ISO-639-3 language codes using macro-languages in case when available. Naturally, training data do not include sentences from Tatoeba and neither from the popular WMT testsets to allow a fair comparison to other models that have been evaluated using those data sets.

Here, we propose an open challenge and the idea is to encourage people to develop machine translation in real-world cases for many languages. The most important point is to get away from artificial setups that only simulate low-resource scenarios or zero-shot translations. A lot of research is tested with multi-parallel data sets and high resource languages using data sets such as WIT[3] (Cettolo et al., 2012) or Europarl (Koehn, 2005) simply reducing or taking away one language pair for arguing about the capabilities of learning translation with little or without explicit training data for the language pair in question (see, e.g., Firat et al. (2016a,b); Ha et al. (2016); Lakew et al. (2018)). Such a setup is, however, not realistic and most probably over-estimates the ability of transfer learning making claims that do not necessarily carry over towards real-world tasks.

In the set we provide here we, instead, include all available data from the collection without removing anything. In this way, the data refers to a diverse and skewed collection, which reflects the real situation we need to work with and many low-resource languages are only represented by noisy or very unrelated training data. Zero-shot scenarios are only tested if no data is available in any of the sub-corpora. More details about the data compilation and releases will be given below.

Tatoeba is, admittedly, a rather easy test set in general but it includes a wide variety of languages and makes it easy to get started with rather encouraging results even for lesser resourced languages. The release also includes medium and high resource settings and allows a wide range of experiments with all supported language pairs including studies of transfer learning and pivot-based methods.

2 Data releases

The current release includes over 500GB of compressed data for 2,961 language pairs covering 555 languages. The data sets are released per language

[1] http://opus.nlpl.eu/
[2] https://tatoeba.org/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1174–1182
Online, November 19–20, 2020 © 2020 Association for Computational Linguistics

pair with the following structure, using deu-eng as an example (see Figure 1).

```
data/deu-eng/
data/deu-eng/train.src.gz
data/deu-eng/train.trg.gz
data/deu-eng/train.id.gz
data/deu-eng/dev.id
data/deu-eng/dev.src
data/deu-eng/dev.trg
data/deu-eng/test.src
data/deu-eng/test.trg
data/deu-eng/test.id
```

Figure 1: Released data packages: training data, development data and test data. Language labels are stored in ID files that also contain the name of the source corpus for the training data sets.

Files with the extension *.src* refer to sentences in the source language (*deu* in this case) and files with extension *.trg* contain sentences in the target language (*eng* here). File with extension *.id* include the ISO-639-3 language labels with possibly extensions about the orthographic script (more information below). In the *.id* file for the training data there are also labels for the OPUS corpus the sentences come from. We include the entire collection available from OPUS with data from the following corpora: ada83, Bianet, bible-uedin, Books, CAPES, DGT, DOGC, ECB, EhuHac, EiTB-ParCC, Elhuyar, EMEA, EUbookshop, EUconst, Europarl, Finlex, skmo, giga-fren, GlobalVoices, GNOME, hrenWaC, infopankki, JRC-Acquis, JW300, KDE4, KDEdoc, komi, MBS, memat, MontenegrinSubs, MultiParaCrawl, MultiUN, News-Commentary, OpusPublik, OpenOffice, OpenSubtitles, ParaCrawl, PHP, QED, RF, sardware, SciELO, SETIMES, SPC, Tanzil, TED2013, TedTalks, TEP, TildeMODEL, Ubuntu, UN, UNPC, wikimedia, Wikipedia, WikiSource, XhosaNavy.

The data sets are compiled from the pre-aligned bitexts but further cleaned in various ways. First of all, we remove non-printable characters and strings that violate Unicode encoding principles using regular expressions and a recoding trick using the forced encoding mode of *recode* (v3.7), a popular character conversion tool.[3] Furthermore, we also de-escape special characters (like '&' encoded as '&') that may appear in some of the corpora. For that, we apply the tools from Moses (Koehn et al., 2007). Finally, we also apply automatic language identification to remove additional noise from the data. We use the compact language detect library (CLD2) through its Python bindings[4] and a Python library for converting between different ISO-639 standards.[5] CLD2 supports 172 languages and we use the options for "best effort" and apply the assumed language from the original data as the "hint language code". For unsupported languages, we remove all examples that are detected to be English as this is a common problem in some corpora where English texts appear in various places (e.g. untranslated text in localization data of community efforts). In all cases, we only rely on the detected language if it is flagged as reliable by the software.

All corpus data and sub-languages are merged and shuffled using terashuf[6] that is capable to efficiently shuffle large data sets. But we keep track of the original data set and provide labels to recognize the origin. In this way, it is possible to restrict training to specific subsets of the data to improve domain match or to reduce noise. The entire procedure of compiling the Tatoeba Challenge data sets is available from the project repository at https://github.com/Helsinki-NLP/Tatoeba-Challenge .

The largest data set (English-French) contains over 180 million aligned sentence pairs and 173 language pairs are covered by over 10 million sentence pairs in our collection. Altogether, there are almost bilingual 3,000 data sets and we plan regular updates to improve the coverage. Below, we give some more details about the language labels, test sets and monolingual data sets that we include in the package as well.

2.1 Language labels and scripts

We label all data sets with standardized language codes using three-letter codes from ISO-639-3. The labels are converted from the original OPUS language IDs (which roughly follow ISO-639-1 codes but also include various non-standard IDs) and information about the writing system (or script) is automatically assigned using Unicode regular expressions and counting letters from specific script character properties. For the scripts we use four-letter codes from ISO-15924 and attach them to the three-letter language codes defined in ISO-639-3. Only the most frequently present script in a string is shown. Mixed content may appear but is not marked specially. Note that the code Zyyy

[3] https://github.com/pinard/Recode

[4] https://pypi.org/project/pycld2/

[5] https://pypi.org/project/iso-639/

[6] https://github.com/alexandres/terashuf

refers to common characters that cannot be used to distinguish scripts. The information about the script is not added if there is only one script in that language and no other scripts are detected in any of the strings. If there is a default script among several alternatives then this particular script is not shown either. Note that the assignment is done fully automatically and no corrections have been made. Three example label sets are given below using the macro-languages Chinese (zho), Serbo-Croatian (hbs) and Japanese (jpn) that can use character from different scripts:

Chinese: cjy_Hans, cjy_Hant, cmn, cmn_Bopo, cmn_Hans, cmn_Hant, cmn_Latn, gan, lzh, lzh_Bopo, lzh_Hang, lzh_Hani, lzh_Hans, lzh_Hira, lzh_Kana, lzh_Yiii, nan_Hani, nan_Latn, wuu, wuu_Bopo, wuu_Hang, wuu_Hani, wuu_Hira, yue_Hans, yue_Hant, yue_Latn

Japanese: jpn, jpn_Hani, jpn_Hira, jpn_Kana, jpn_Latn

Serbo-Croatian: bos_Latn, hrv, srp_Cyrl, srp_Latn

This demonstrates that a data set may include examples from various sub-languages if they exist (e.g. Bosnian, Croatian and Serbian in the Serbo-Croatian case) or language IDs with script extensions that show the dominating script in the corresponding string (e.g. Cyrl for Cyrillic or Latn for Latin script). Those labels can be used to separate the data sets, to test sub-languages or specific scripts only or to remove some noise (like the examples that are tagged with the Latin script (Latn) in the Japanese data set. Note that script detection can also fail in which the corresponding code is missing or potentially wrong. For example, the detection of traditional (Hant) och simplified Chinese (Hans) can be ambiguous and encoding noise can have an effect on the detection.

We also release the tools that we developed for converting and standardizing OPUS IDs and also the tools that detect scripts and variants of writing systems. The package is available from github[7] and can be installed from CPAN[8].

2.2 Multiple reference translations

Test and development data are taken from a shuffled version of Tatoeba. All translation alternatives are included in the data set to obtain the best coverage of languages in the collection. Development and test sets are disjoint in the sense that they do not include identical source-target language sentence pairs. However, there can be identical source sentences or identical target sentences in both sets, which are not linked to the same translations. Similarly, there can be identical source or target sentences in one of the sets, for example the test set, with different translations. In Figure 2, you can see examples from the Esperanto-Ladino test set.

epo	ladLatn
u vi estas en Berlino?	Estash en Berlin?
u vi estas en Berlino?	Vos estash en Berlin?
u vi estas en Berlino?	Vozotras estash en Berlin?
La hundo estas nigra.	El perro es preto.
La hundo nigras.	El perro es preto.

Figure 2: Examples of test sentences with multiple reference translations taken from the Esperanto-Ladino test set.

The test data could have been organized as multi-reference data sets but this would require to provide different sets in both translation directions. Removing alternative translations is also not a good option as this would take away a lot of relevant data. Hence, we decided to provide the data sets as they are, which implicitly creates multi-reference test sets but with the wrong normalization.

2.3 Monolingual data

In addition to the parallel data sets we also provide monolingual data that can be used for unsupervised methods or data augmentation approaches such as back-translation. For that purpose, we extract public data from Wikimedia including source from Wikpedia, Wikibooks, Wikinews, Wikiquote and Wikisource. We extract sentences from data dumps provided in JSON format[9] and process them with jq,[10] a lightweight JSON processing tool. We apply the same cleaning steps as we do for the OPUS bitexts including language identification and convert language IDs to ISO-639-3 as before. Sentence boundaries are detected using UDPipe (Straka et al., 2016) with models trained on universal dependency treebanks v 2.4 and the Moses sentence splitter with language-specific non-breaking prefixes if available. We preserve document boundaries and do not shuffle the data to enable experiments with discourse-aware models. The data sets are released along with the rest of the Tatoeba challenge data.

3 The translation challenge

The main challenge is to develop translation models and to test them with the given test data from

[7] https://github.com/Helsinki-NLP/LanguageCodes

[8] https://metacpan.org/pod/ISO::639::3 and https://metacpan.org/pod/ISO::639::5

[9] https://dumps.wikimedia.org/other/cirrussearch/current

[10] https://stedolan.github.io/jq/

Tatoeba. The focus is on low-resource languages and to push their coverage and translation quality. Resources for high-resource are also provided and can be used as well for translation modeling of those languages and for knowledge transfer to less resourced languages. Note that not all language pairs have sufficient data sets for test, development (*dev*) and training (*train*) data. Hence, we divided the Tatoeba challenge data into various subsets based on the size of the training data available.

high-resource settings: 298 language pairs with training data of at least one million training examples (aligned sentence pairs), we further split into language pairs with more than 10 million training examples (173 language pairs) and other language pairs with data sets below the size of 10 million examples

medium-sized resource settings: 97 language pairs with more than 100,000 and less than 1 million training examples

low-resource settings: 87 language pairs with less than 100,000 training examples, we further distinguish between language pairs with more than 10,000 training examples (63) and language pairs below 10,000 training examples (24)

zero-shot translation: language pairs with no training data (40 in the current data set)

For all those 522 selected language pairs, the data set provides at least 200 sentences per test set. 101 of them involves English as one of the languages. 288 test sets contain more than 1,000 sentence pairs of which only 68 include English. Note, that everything below 1,000 sentences is probably not very reliable as a proper test set but we decided to release smaller test sets as an initial benchmark to trigger further development even for extremely under-resourced language pairs. We also decided to use very low thresholds for the division into low-resource languages. Having 10,000 training examples or less is very realistic for many real-world examples and we want to encourage the work on such cases in particular.

The maximum size of test sets in our collection is 10,000 sentence pairs, which is available for 76 language pairs. The test size is reduced to 5,000 if there is less than 20,000 sentence pairs in Tatoeba (119 data sets). The remaining sentences are released as disjoint validation data. For 48 Tatoeba language pairs with less than 10,000 sentence pairs, we keep 2,500 for the test set and the rest for validation and for 78 Tateoba language pairs with less than 5,000 sentence pairs we keep 1,000 for validation and the rest for testing. Finally, for language pairs with less than 2,000 sentences in Tatoeba we skip validation data and use everything for test purposes.

Test and validation data are strictly disjoint and none of the examples from Tatoeba are explicitly included in the training data. However, as it is common in realistic cases, there is a natural chance for a certain overlap between those data sets. Figure 3 plots the percentage of sentence pairs in test and validation sets that can also be found in the corresponding training data we release. The average proportion is rather low around 5.5% for both with a median percentage of 2.3% and 2.9% for test and validation data, respectively. There is one clear outlier with a very high proportion of over 55% overlap and that is Danish–English for some reason that is not entirely clear to us. Otherwise, the values are well below that ratio.

4 The data challenge

The most important ingredient for improved translation quality is data. It is not only about training data but very much also about appropriate test data that can help to push the development of transfer models and other ideas of handling low-resource settings. Therefore, another challenge we want to open here is the increase of the coverage of test sets for low-resource languages. Our strategy is to organize the extension of the benchmarks directly through the Tatoeba initiative. Users who would like to contribute to further MT benchmark development are asked to register for the open service provided by Tatoeba and to upload new translations in the languages of interest. From our side, we will continuously update our challenge data set to include the latest data releases coming from Tatoeba including new language pairs and extended data sets for existing language pairs. We will make sure that the new test sets do not overlap with any released development data from previous revisions to enable fair comparisons of old models with new benchmarks. The extended test and validation data sets will be released as new packages and old revisions will be kept for replicability of existing

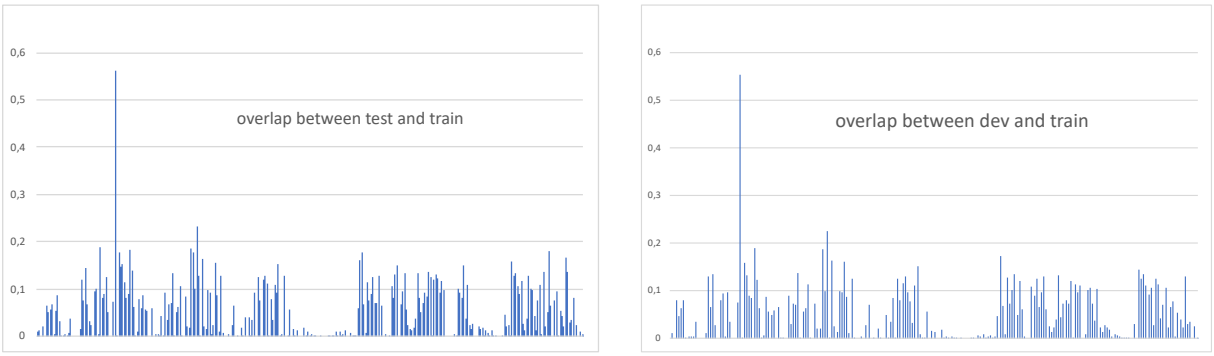

Figure 3: Overlap between test and validation (dev) data and the training data: Proportion of sentence pairs that exist in the training data for all data sets above 1,000 sentence pairs.

scores.

In order to provide information about language pairs in need, we provide a list of data sets with less than 1,000 examples per language pair. In the current release, this refers to 2,375 language pairs. 2,141 language pairs have less than 200 translation units and are, therefore, not included in the released benchmark test set. Furthermore, we also provide a list of languages for which we release training data coupled with English but no test data is available from Tatoeba. Currently, this relates to 246 languages.

We encourage users to especially contribute translations for those data sets in order to improve the language coverage even further. We hope to trigger a grass-root development that can signi - cantly boost the availability of development and test sets as one of the crucial elements for pushing NMT development in the corresponding languages.

Finally, we also encourage to incorporate other test sets besides of the Tatoeba data. Currently, we also test with WMT news test sets for the language pairs that are covered by the released development and test sets over the years of the news translation campaign. Contributions and links can be provided through the repository management interface at github.

5 How to participate

The goal of the data release is to enable a straightforward setup for machine translation development. Everyone interested is free to use the data for their own development. A leader board for individual language pairs will be maintained. Furthermore, we also intend to make models available that are listed in the challenge. This does not only support replicability but also provides a new unique resource of pre-trained models that can be integrated in real-world applications or can be used in further research, unrelated downstream tasks or as a starting point for subsequent ne-tuning and domain adaptation. A large number of models is already available from our side providing baselines for a large portion of the data set. More details will be provided below.

For participation, there are certain rules that apply:

- Do not use any development or test data for training (*dev* can be used for validation during training as an early stopping criterion).
- Only use the provided training data for training models with comparable results in constrained settings. Any combination of language pairs is ne or backtranslation of sentences included in training data for any language pair is allowed, too. That means that additional data sets, parallel or monolingual, are not allowed for of cial models to be compared with others.
- Unconstrained models may also be trained and can be reported as a separate category. Using pre-trained language or translation models fall into the unconstrained category. Make sure that the pre-trained model does not include Tatoeba data that we reserve for testing.
- We encourage to release models openly to ensure replicability and re-use of pre-trained models. If you want to enter the of cial leader board you have to make your model available including instructions on how to use them.

6 Baseline Models

Along with the data, we also release baseline models that we train with state-of-the-art trans-

former models (Vaswani et al., 2017) using Marian-NMT,[11] a stable production-ready NMT toolbox with ef cient training and decoding capabilities (Junczys-Dowmunt et al., 2018). We apply a common setup with 6 self-attentive layers in both, the encoder and decoder network using 8 attention heads in each layer. The hyper-parameters follow the general recommendations given in the documentation of the software[12]. The training procedures follow the strategy implemented in OPUS-MT (Tiedemann and Thottingal, 2020) and detailed instructions are available from github[13].

We train a selection of models on v100 GPUs with early-stopping after 10 iterations of dropping validation perplexities. We use SentencePiece (Kudo and Richardson, 2018) for the segmentation into subword units and apply a shared vocabulary of a maximum of 65,000 items. Language label tokens in the spirit of Johnson et al. (2017) are used in case of multiple language variants or scripts in the target language. Models for over 400 language pairs are currently available and we refer the reader to the website with the latest results. For illustration, we provide some example scores below in Table 1 using automatic evaluation based on chrF2 and BLEU computed using sacrebleu (Post, 2018). The actual translations are also available for each model and the distribution comes along with the log les from the training process and all necessary data les such as the SentencePiece models and vocabularies.

language pair	chrF2	BLEU
aze-eng	0.490	31.9
bel-eng	0.268	10.0
cat-eng	0.668	50.2
eng-epo	0.577	35.6
eng-glg	0.593	37.8
eng-hye	0.404	16.6
eng-ilo	0.569	30.8
eng-run	0.436	10.4

Table 1: Translations scores from baseline models trained for a selection of medium-size language pairs (according to our classi cation) tested on the provided Tatoeba benchmark. We show here models that include English and score above 10 BLEU.

[11] https://marian-nmt.github.io

[12] https://github.com/marian-nmt/marian-examples/tree/master/transformer

[13] https://github.com/Helsinki-NLP/OPUS-MT-train/blob/master/doc/TatoebaChallenge.md

7 Multilingual Models

One of the most interesting questions is the ability of multilingual models to push the performance of low-resource machine translation. The Tatoeba translation challenge provides a perfect testbed for systematic studies on the effect of transfer learning across various subsets of language pairs. We already started various experiments with a number of multilingual translation models that we evaluate on the given benchmarks. In our current work, we focus on models that include languages in established groups and for that we facilitate the ISO-639-5 standard. This standard de nes a hierarchy of language groups and we map our data sets accordingly to start new models that cover those sets. As an example, we look at the task of Belorussian-English translation that has been included in the previous section as well. Table 2 summarizes the results of our current models sorted by chrF2 scores.

model	chr-F2	BLEU
sla-eng/opus4m	0.610	42.7
sla-eng/opus2m	0.609	42.5
sla-eng/opus1m	0.599	41.7
ine-eng/opus2m	0.597	42.2
ine-eng/opus4m	0.597	41.7
ine-eng/opus1m	0.588	41.0
zle-eng/opus4m	0.573	38.7
zle-eng/opus2m	0.569	38.3
mul-eng/opus1m	0.550	37.0
mul-eng/opus2m	0.549	36.8
zle-eng/opus1m	0.543	35.4
ine-ine/opus1m	0.512	31.8
bel-eng/opus	0.268	10.0

Table 2: Translation results of the Belorussian-English test set using various multilingual translation models compared to the baseline bilingual model (shown at the bottom). opusXm refers to sampled data sets that include X million sentences per language pair.

The models focus on different levels of relatedness of the languages and range from East Slavic Languages (zle), Slavic languages (sla) to the language family of Indo-European languages (ine) and the set that contains all languages (mul). Each model is trained on sampled data set in order to balance between different languages. The smallest training sets are based on data that are sampled to include a maximum of one million sentence per language pair (opus1m). We use both, down-sampling and up-sampling. The latter is done by simply

multiplying the existing data until the threshold is reached. We also set a threshold of 50 for the maximum of repeating the same data in order to avoid over-representing small noisy data. The one-million models are trained first and form the basis of larger models. We continue training with data sets sampled to two million before increasing to four million sentence pairs.

The Table shows some interesting patterns. First of all, we can clearly see a big push in performance when adding related languages to the training data. This is certainly expected especially in the case of Belorussian that is closely related to higher-resource-languages such as Russian and Ukrainian. Interesting is that the East Slavic language group is not the best performing model even though it includes those two related languages. The additional information from other Slavic languages pushes the performance beyond their level quite significantly. Certainly, those models will see more data and this may cause the difference. The 'sla-eng' model covers 13 source languages whereas 'zle' only 5. Also interesting to see is that the Indo-European language model fairs quite well despite the enormous language coverage that this model has to cope with. On the other hand, the big 'mul' translation model does not manage to create the same performance and the limits of the standard model with such a massive setup become apparent. Training those models becomes also extremely expensive and slow and we did not manage to start the 4-million-sentence model.

Currently, we look into the various models we train and many other interesting patterns can be seen. We will leave a careful analyses to future work and also encourage the community to explore this field further using the given collection and benchmark. Updates about models and scores will be published on the website and we would also encourage more qualitative studies that we were not able to do yet.

8 Zero-shot and few-shot translation

Finally, we have a quick look at zero-shot and few-shot translation tasks. Table 3 shows results for Awadhi-English translation, one of the test sets for which no training data is available. Awadhi is an Eastern Hindi language in the Indo-Iranian branch of the Indo-European language family.[14]

[14]We use ISO639-3 and ISO639-5 standards for names and codes of languages and language groups.

model	chr-F2	BLEU
ine-eng/opus1m	0.285	10.0
mul-eng/opus1m	0.257	9.4
inc-eng/opus1m	0.217	6.8
iir-eng/opus1m	0.214	7.9
ine-ine/opus1m	0.201	2.4
tatoeba-zero/opus	0.042	0.1

Table 3: Translation results of the Awadhi-English test set using multilingual translation models.

The table shows that a naive approach of throwing all languages that are part of zero-shot language pairs into one global multilingual model (tatoeba-zero) does not work well. This is probably not very surprising. Another interesting observation is that a symmetric multilingual model with Indo-European languages on both sides (ine-ine) also underperforms compared to other multilingual models that only translate into English. Once again, the Indo-European-language-family to English model performs quite well. Note that the performance purely comes from overlaps with related languages as no Awadhi language data is available during training. The performance is still very poor and needs to be taken with a grain of salt. They demonstrate, however, the challenges one faces with realistic cases of zero-shot translation.

In Table 4, we illustrate another case that could be described as a realistic few-shot translation task. Our collection comes with 3,613 training examples for the translation between English and Faroese. The table shows our current results in this task using multilingual models that translate from English to language groups including the Scandinavian language in question.

model	chr-F2	BLEU
eng-gem/opus	0.318	9.4
gem-gem/opus	0.312	7.0
eng-gmq/opus	0.311	7.0
eng-ine/opus	0.281	6.3
eng-mul/opus	0.280	5.7
ine-ine/opus	0.276	5.9
tatoeba-zero/opus	0.042	0.1

Table 4: Translation results of the English-Faroese test set with different multilingual NMT models.

Again, we can see that the naive tatoeba-zero model is the worst. The symmetric Indo-European model performs better but the English-Germanic

model gives the best performance, which is still very low and not satisfactory for real-world applications. Once again, the example demonstrates the challenge that is posed by extremely low-resource scenarios and we hope that the data set we provide will trigger additional fascinating studies on a large variety of interesting cases.

9 Comparison to the WMT news task

Finally, we also include a quick comparison to the WMT news translation task, see Table 5. Note that we did not perform any optimization for that task, did not use any in-domain back-translations and did not run fine-tuning in the news domain. We only give results for English–German (in both directions) for the 2019 test data to give an impression about the released baseline models.

English – German		
model	BLEU	chr-F2
eng-deu	42.4	0.664
eng-gmw	35.9	0.616
eng-gem	35.0	0.613
eng-ine	26.6	0.554
eng-mul	21.0	0.512
WMT best	44.9	–
German – English		
model	BLEU	chr-F2
deu-eng	40.5	0.645
gmw-eng	36.6	0.615
gem-eng	37.2	0.618
ine-eng	31.7	0.571
mul-eng	27.0	0.529
WMT best	42.8	–

Table 5: Translation results of baseline models or English–German news translation from WMT 2019 using bilingual and multilingual Tatoeba baseline models. The BLEU scores are also compared to the best score that is currently available from http://matrix.statmt.org/matrix – retrieved on October 4, 2020.

The results demonstrate that the models can achieve high quality even on a domain they are not optimized for. The best scores in the German–English case are close to the top performing model registered for this task even though the comparison is not fair for various reasons. The purpose is anyway not to provide state-of-the-art models for the news translation task but baseline models for the Tatoeba case and in future work we will also explore the use of our models as the basis for systems that can be developed for other benchmarks and applications. In the example we can also see that multilingual models significantly lag behind bilingual ones in high-resource cases. Each increase of the language coverage (except for the move from West Germanic languages (gmw) to Germanic languages (gem) in the German–English case) leads to a drop in performance but note that those multilingual models are not fine-tuned for translating from and to German.

10 Conclusions

This paper presents a new comprehensive data set and benchmark for machine translation that covers roughly 3,000 language pairs and over 500 languages and language variants. We provide training and test data that can be used to explore realistic low-resource scenarios and zero-shot machine translation. The data set is carefully annotated with standardized language labels including variations in scripts and with information about the original source. We also release baseline models and results and encourage the community to contribute to the data set and machine translation development. All tools for data preparation and training bilingual as well as multilingual translation models are provided as open source packages on github. We are looking forward to new models, extended test sets and a better coverage of the World's languages.

Acknowledgements

This work is supported by the FoTran project (grant agreement No 771113), funded by the European Research Council (ERC) and the MeMAD project (grant agreement No 780069) under the European Union's Horizon 2020 research and innovation program. We would also like to acknowledge the support of the CSC IT Center for Science, Finland, for computational resources.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine

translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel M. Lakew, Marcello Federico, Matteo Negri, and Marco Turchi. 2018. Multilingual neural machine translation for low-resource languages. *IJCoL - Italian Journal of Computational Linguistics*, 4(1). Emerging Topics at the Fourth Italian Conference on Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-u files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, Istanbul, Turkey.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.